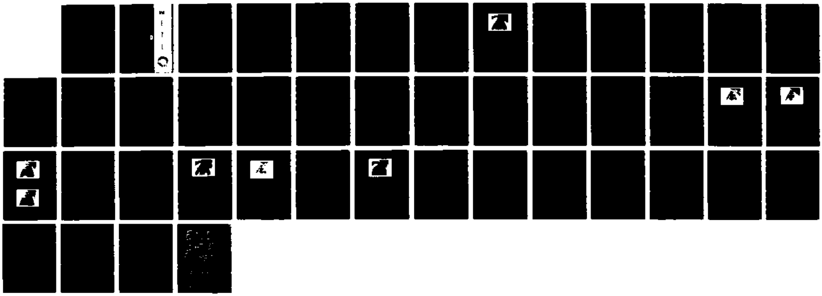
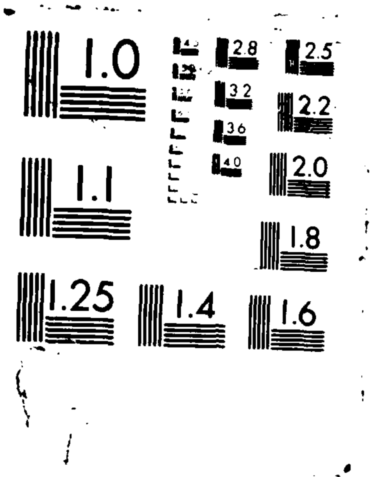
AD-A190 345 AN EXPERT VISION SYSTEM FOR AUTONOMOUS LAND VEHICLE ROAD FOLLOWING..(U) MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION RESEARCH S J DICKINSON ET AL. JAN 88 1/1

UNCLASSIFIED CAR-TR-330 ETL-0489 DACA76-84-C-0004 F/G 12/9 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AD-A190 345

(4)

ETL-0489

An expert vision system for autonomous land vehicle road following

Sven J. Dickinson
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, Maryland 20742

January 1988

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for:

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

88

E T L

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

A190 265

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS: N/A

2a SECURITY CLASSIFICATION AUTHORITY: N/A

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE: N/A

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S): CAR-TR-330; CS-TR-1932

5. MONITORING ORGANIZATION REPORT NUMBER(S): ETL-0489

6a. NAME OF PERFORMING ORGANIZATION: University of Maryland

6b. OFFICE SYMBOL *(If applicable)*: N/A

7a. NAME OF MONITORING ORGANIZATION: U.S. Army Engineer Topographic Laboratories

6c. ADDRESS *(City, State, and ZIP Code)*:
Center for Automation Research
College Park, MD 20742-3411

7b ADDRESS *(City, State, and ZIP Code)*:
Fort Belvoir, VA 22060-5546

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: Defense Advanced Research Projects Agency

8b OFFICE SYMBOL *(If applicable)*: ISTO

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: DACA76-84-C-0004

8c. ADDRESS *(City, State, and ZIP Code)*:
1400 Wilson Blvd.
Arlington, VA 22209

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
62301E			

11 TITLE *(Include Security Classification)*: An Expert Vision System for Autonomous Land Vehicle Road Following

12 PERSONAL AUTHOR(S): Sven J. Dickinson and Larry S. Davis

13a TYPE OF REPORT: Technical

13b TIME COVERED: FROM ________ TO N/A

14. DATE OF REPORT *(Year, Month, Day)*: January 1988

15. PAGE COUNT: 41

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*:
Image Understanding; Structured Blackboard; Vision-Based Navigation; Computer Vision Processing

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A production system model of problem solving is applied to the design of a vision system by which an autonomous land vehicle (ALV) navigates roads. The ALV vision task consists of hypothesizing objects in a scene model and verifying these hypotheses using the vehicle's sensors. Object hypothesis generation is based on the local navigation task, an a priori road map, and the contents of the scene model. Verification of an object hypothesis involves directing the sensors toward the expected location of the object, collecting evidence in support of the object, and reasoning about the evidence. Constructing the scene model consists of building a semantic network of object frames exhibiting component, spatial, and inheritance relationships. The control structure is provided by a set of communicating production systems implementing a structured blackboard; each production system contains rules for defining the attributes of a particular class of object frame. The combination of production system and object oriented programming techniques results in a flexible control structure able to accommodate new object classes, reasoning strategies, vehicle sensors, and image analysis techniques.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL: Rosalene Holecheck

22b TELEPHONE *(Include Area Code)*: (202) 355-2767

22c. OFFICE SYMBOL: ETL-RI-T

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Acknowledgements

Many people deserve thanks for their contribution to this work. In particular, the authors would like to thank Jacqueline LeMoigne and Rand Waltzman for insightful discussions at the early stages of the project. David Harwood and Sergio Antoy offered helpful suggestions during the design phase, while Todd Kushner and T.K. Siddalingaiah provided hardware support in the implementation. The authors would also like to thank Suzanne Stevenson and Azriel Rosenfeld for their excellent comments on earlier drafts of this report.

Contents

List of Figures

Figure 1: A Typical ALV Road Image

1 Introduction

The development of an Autonomous Land Vehicle (ALV) involves the development of computer vision techniques by which a vehicle can autonomously navigate itself through the environment. Although the goals for the ALV are broad, including both on- and off-road navigation, the work presented here is primarily concerned with the road following task. Figure 1 presents a view as seen from a camera mounted on top of the ALV. From images such as these, the ALV vision system constructs a model of the environment; this scene model contains the objects visually identified by the ALV. Based on this collection of objects, the vehicle plans a course and moves through the environment.

For the road following task, the scene model contains either objects that represent the road or objects from which the location of the road can be deduced. Obviously, the direct detection of a patch of road would be most useful; however, in the event that the ALV vision system cannot directly identify the road, the detection of other objects may suggest the location of the road. For example, telephone poles and ditches often run parallel to the road; their presence may thus provide clues as to its location and direction. In certain cases, major landmarks contained in a road map such as buildings may be used to infer the road location; however, such information is more useful in registering the vehicle to some absolute location.

Faced with the task of building a scene model, the ALV vision system must decide which of the above objects should be sought in order to locate the road. Hence, the first

step in constructing the scene model is the joint decision of what object to search for in the world and where to search for it. The second step performed by the ALV vision system is to gather evidence confirming the existence of the specified object at the hypothesized location. The third and final step involves reasoning about the evidence. If the confirming evidence is sufficient, the object is inserted into the scene model; otherwise, the object may be hypothesized elsewhere or a different object hypothesized.

Construction of the scene model is complex. The selection of which object to track depends on the navigation goals of the ALV, the history of object tracking, the contents of the scene model, and information from the road map. Verifying the existence of an object requires directing vehicle sensors towards the object, fusing data from different sensors, and selecting algorithms for image analysis. Methods for performing all these tasks are continually evolving as the road following task becomes better understood. New objects must be tracked by the ALV, new sensors are available to track objects, and new image processing techniques are identified for sensor image feature extraction. The successful evolution of an ALV vision system hinges on the ability of its control structure and knowledge representation schemes to accommodate these changes.

We propose the design of a system for constructing an ALV scene model, offering a flexible control structure able to accommodate new strategies for object tracking, sensor selection, and feature extraction. The goal of the system is to provide a flexible tool for the development of ALV road following software. The design is based on concepts described in [Hanson & Riseman], but offers a unique implementation based on a set of communicating production systems.

The scene model is constructed as a network of frames, each frame corresponding to a class of objects and encapsulating the relevant information pertaining to that class. The control structure used to build the network is based on a system of communicating production systems implementing a structured blackboard. Each region of the blackboard corresponds to a particular class of frames and contains the rules which define the attributes of the class. The system promotes modularity and maintainability through a structured object representation and a structured control scheme.

In the following section, we provide an overview of the system before exploring in detail the representation and control schemes. In Section 3 we discuss object modeling and describe the object classes currently available in the system, while in Section 4 we discuss the scene model network. Sections 5 and 6 discuss the control strategies involved in the selection and verification of objects, respectively. We conclude with a series of results demonstrating the system's capabilities, and discuss the evolution of the system.

2 System Overview

The task of building a scene model for the ALV consists of two major subtasks:

1. deciding what object to look for and where to look for it;
2. verifying that the object exists in the world.

These two functions are performed by the Scene Model Planner (the Planner) and Scene Model Verifier (the Verifier), respectively; together, they form the Scene Model Builder (the Builder). The data flow diagram for the Builder is presented in Figure 2. The Planner, in addition to interpreting and updating the scene model, is aware of the local navigation task and initiates queries to the a priori road map. (The local navigation task is a function of the global navigation task and the location of the vehicle; for example, a local navigation task might be to follow the road for 100 meters, or turn right at the first intersection after a certain landmark is identified. The road map contains a priori information about the ALV environment, including the approximate locations, sizes, and compositions of roads, intersections, and landmarks.) The Verifier controls the movement of the sensors and acquires the sensor image data. In a hypothesize-and-test paradigm, the Planner sends object hypotheses to the Verifier, while the Verifier returns verified objects to the Planner.

The dataflow of the Builder proceeds as follows. The Planner first determines the scene model requirements of the local navigation task; for example, following a straight road requires that the left and right road boundaries be contained in the scene model. Next, the Planner looks at the road map and the partial scene model and decides what objects may be useful in locating the road; for example, it might decide that a road patch, a ditch, or even a row of telephone poles is sufficient to define a road boundary. The Planner then decides the type and expected location of the object to be tracked, hypothesizes the object, and passes the hypothesis to the Verifier.

The Verifier attempts to verify the hypothesis by directing the vehicle's image sensors towards the expected location of the object. The object is then located in the sensor images and its image location is mapped to a 3-D location based on a fixed point of reference. The confidence with which the object is found becomes a measure of its verifiability. Once the confidence is defined, the hypothesis is returned to the Planner for inspection. If the object is deemed sufficiently verified, it is added to the scene model. Otherwise, the Planner determines the next course of action; for example, the object may be hypothesized in a different location, or a new object hypothesized.

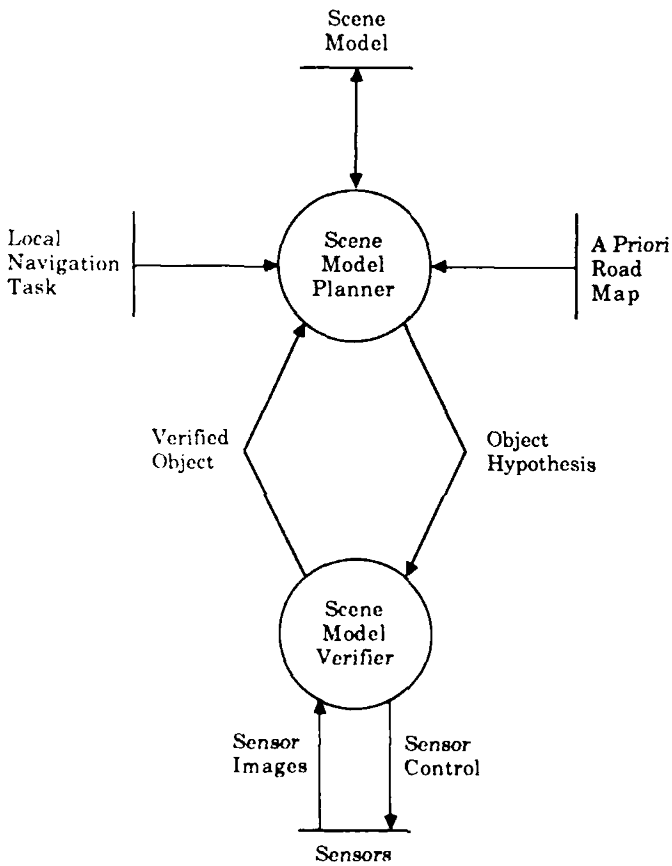


Figure 2: The Scene Model Builder Dataflow

3 Modeling World Objects

Objects in the ALV scene model exhibit the following relationships:

- *Component Relationships.* For example, an intersection is made up of four connecting roads, stop lights, etc. These component objects, in turn, can be decomposed into their component objects, e.g. a road may be defined as a pair of left and right segments, each edge representing the border between road and shoulder, or shoulder and background. Primitive objects such as lines and surfaces extracted from an image cannot be decomposed.

- *Spatial Relationships.* For example, telephone poles are often located near the road and run parallel to the road.

- *Property Inheritance.* For example, a three-dimensional line segment may be defined by a pair of endpoints. The edge separating the road surface from the shoulder is a specialization of a three-dimensional line segment; thus, in addition to its properties specific to a road edge, it inherits the endpoint properties of the three-dimensional line segment.

To accommodate these relationships, frames have been chosen to model objects [Minsky]. A frame is a data structure containing a set of slots (or attributes) which encapsulate the relevant knowledge pertaining to an object. Slots may contain values, e.g. the width of a road patch, or pointers to related frames, e.g. component and spatially related frames. Property inheritance among frames is accomplished by supplementing a frame's slots with the inherited frame's slots. The following sections describe the object frames defined in the system; the frame attributes are discussed in more detail in Appendix A.

3.1 The Road Patch

A planar ribbon is defined as a pair of facing and parallel three-dimensional line segments. A road patch is a specialization of a planar ribbon, whose three-dimensional line segments represent the left and right features of the road. Thus, a road patch frame inherits the attributes of a planar ribbon. The road patch and planar ribbon frames are depicted in Figure 3. Road patches are oriented and may be connected together to form a piece of road; the front of one road patch may be connected to the back of another. The orientation of a road patch is based on the assigned orientation of the initial road patch; typically, the back end of the initial road patch is closest to the ALV, while the front end is furthest from the ALV. The left and right features, i.e. three-dimensional segments, are oriented looking from the back to the front of the road patch. Figure 4 depicts the vehicle with respect to a series of road patches.

planar-ribbon

attributes:

search-location
search-strategy
back-connected-planar-ribbon
front-connected-planar-ribbon
has-part-left-world-segment
has-part-right-world-segment
expected-width
actual-width
actual-width-confidence
parallelism-confidence
total-confidence

road-patch

attributes:

prior-road-straightness
left-world-segment-type
right-world-segment-type

inherited-frames:

planar-ribbon

Figure 3: The Road Patch/Planar Ribbon Frames

3.2 The Road Patch Segment

A world segment is defined as a three-dimensional line segment. A road patch segment is a specialization of a world segment representing a road feature, i.e. the boundary between the road surface and the shoulder surface or the boundary between the shoulder surface and the vegetation or background. Thus a road patch segment frame inherits the attributes of a world segment. The road patch segment and world segment frames are depicted in Figure 5. Road patch segments are oriented and may be connected together to form a continuous linear feature; the front of one road patch segment may be connected to the back of another. The orientation of a road patch segment is based on the orientation of its parent road patch.

3.3 The Road Patch Camera Segment

A camera segment is defined as a two-dimensional line segment extracted from a camera image. A road patch camera segment is a specialization of a camera segment representing the two-dimensional projection of a three-dimensional road feature. Thus a road patch camera segment frame inherits the attributes of a camera segment frame. The road patch camera segment and camera segment frames are depicted in Figure 6. Road patch camera segments are oriented and may be connected together to form a continuous two-dimensional linear feature; the front of one road patch camera segment may be connected to the back of another. The orientation of a road patch camera segment is based on the orientation of its parent road patch segment.

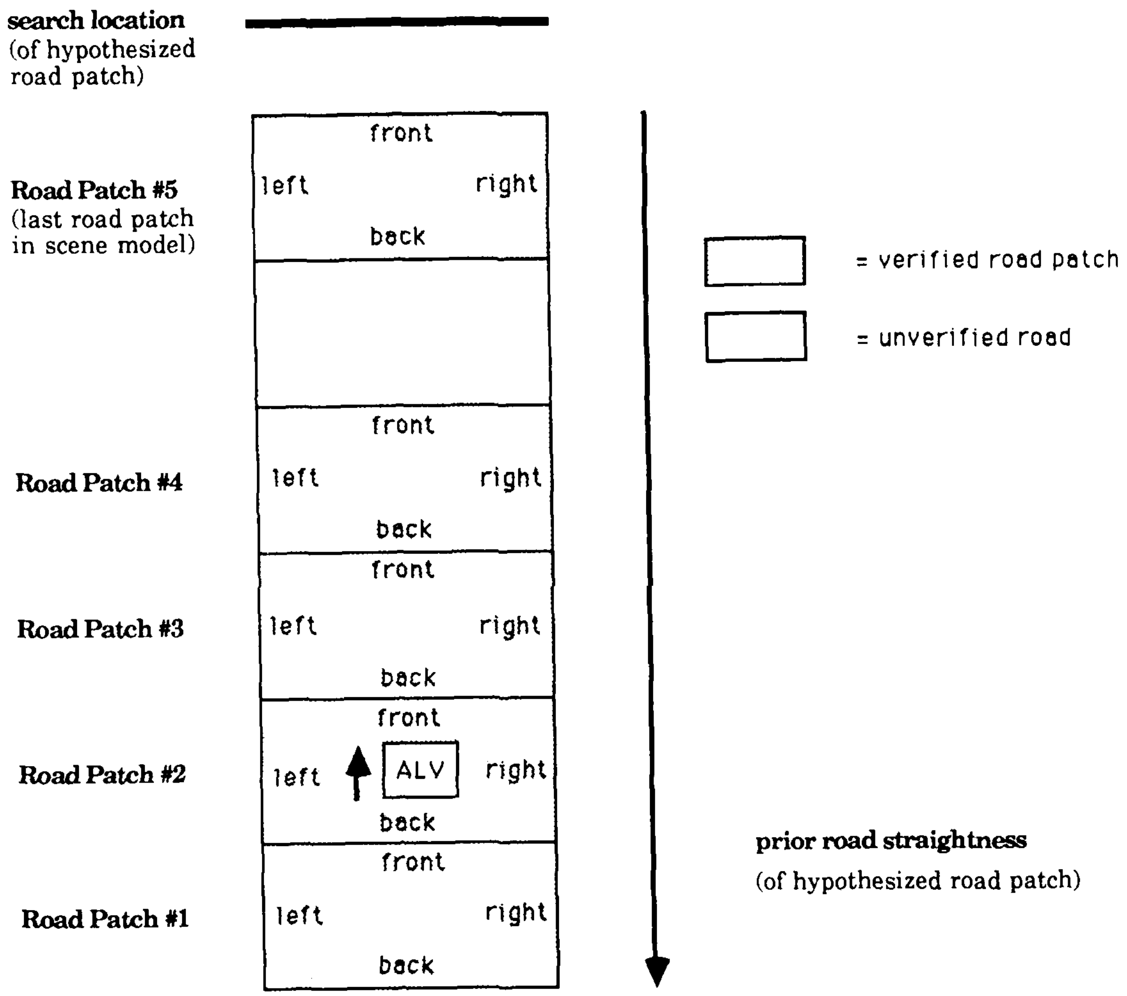


Figure 4: A Series of Road Patches

world-segment

attributes:

search-location
search-strategy
endpoint-A
endpoint-B
endpoint-A-connected-world-segment
endpoint-B-connected-world-segment
part-of-planar-ribbon
has-part-camera-segment
total-confidence

road-patch-segment

attributes:

world-segment-feature
endpoint-A-orientation
endpoint-B-orientation
continuity-confidence

inherited-frames:

world-segment

Figure 5: The Road Patch Segment/World Segment Frames

camera-segment

attributes:

search-window
search-strategy
endpoint-A
endpoint-B
endpoint-A-connected-camera-segment
endpoint-B-connected-camera-segment
part-of-world-segment
camera-image
method-of-extraction
total-confidence

road-patch-camera-segment

attributes:

camera-segment-feature
endpoint-A-orientation
endpoint-B-orientation

inherited-frames:

camera-segment

Figure 6: The Road Patch Camera Segment/Camera Segment Frames

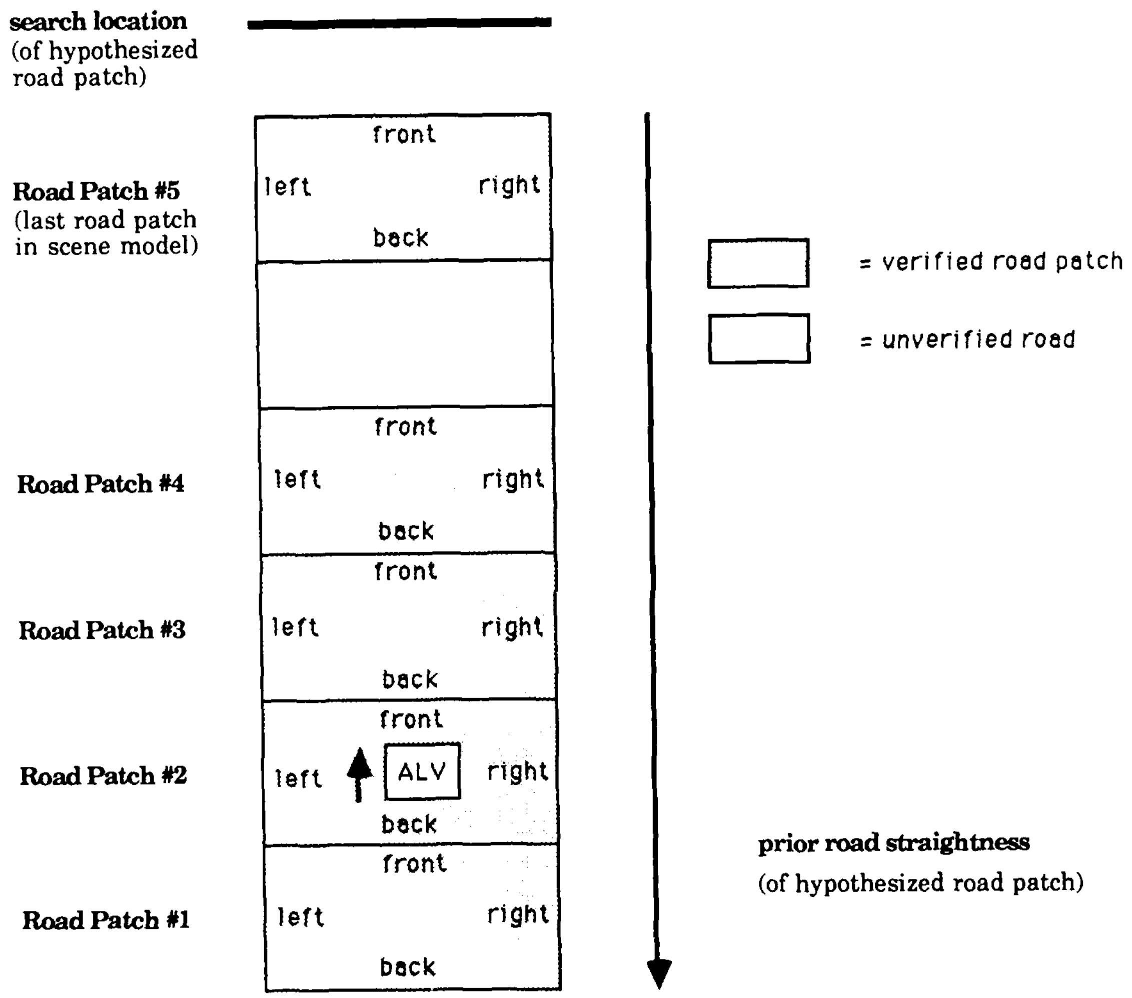


Figure 4: A Series of Road Patches

world-segment

attributes:

search-location
search-strategy
endpoint-A
endpoint-B
endpoint-A-connected-world-segment
endpoint-B-connected-world-segment
part-of-planar-ribbon
has-part-camera-segment
total-confidence

road-patch-segment

attributes:

world-segment-feature
endpoint-A-orientation
endpoint-B-orientation
continuity-confidence

inherited-frames:

world-segment

Figure 5: The Road Patch Segment/World Segment Frames

camera-segment

attributes:

search-window
search-strategy
endpoint-A
endpoint-B
endpoint-A-connected-camera-segment
endpoint-B-connected-camera-segment
part-of-world-segment
camera-image
method-of-extraction
total-confidence

road-patch-camera-segment

attributes:

camera-segment-feature
endpoint-A-orientation
endpoint-B-orientation

inherited-frames:

camera-segment

Figure 6: The Road Patch Camera Segment/Camera Segment Frames

4 The Scene Model

The scene model is central to the ALV vision system. It is accessed by the Planner in determining what object to search for and by the ALV navigator when plotting a course through the scene model objects. As the domain of scene model objects grows larger, so does the variety of requests to the scene model. It is important that the *semantics* of the access commands be stable while the structure of the scene model evolves to accommodate new object classes. Hence, the structure of the scene model is transparent to those modules that access it. As in the case of the scene model objects, the scene model is a frame defining a set of query functions and hiding all implementation details. For example, queries can be made to determine the length of straight road at the end of the scene model, or what road boundary, i.e. road or shoulder edge, has been verified most successfully.

Given the present limited domain of scene model objects, i.e. road patches and their components, a connected graph of road patches serves as an adequate scene model. However, as more object classes are defined, such an implementation may become insufficient; a structure accommodating both spatial and symbolic data would be preferred. Figure 7 depicts a scene model containing three road patches; two of the road patches are connected to each other while the third is disconnected. Note that the links between a road patch and its component road patch segments, and between a road patch segment and its component road patch camera segment, are shown. In addition, the network includes the links between connected road patches, road patch segments, and road patch camera segments. As the domain of objects does not yet include intersecting roads, the current implementation of the scene model is quite simple. The road patches are kept in a list ordered by their occurrence along the road; successive road patches in the scene model are not necessarily connected. When inserted into the scene model, each road patch is "trimmed" so that the front and back endpoints of its component road patch segments are square.

The frames comprising the road patches in Figure 7 can be divided into two layers. The upper layer, including the road patch frame and its two road patch segment component frames, contains objects and their components as they are defined in the world. The lower layer, including the road patch camera segment frames, contains objects as they are defined in the sensor images. This division facilitates the incorporation of new sensors to the vehicle. For example, if we add a range scanner to the ALV, then a road patch segment frame would be given an additional slot defining a component road patch range segment. Aside from the additional attribute, the only impact on the road patch segment frame would be an alteration of the *endpoint A (B)* calculation to include the data from the road patch range segment, and an alteration of the total confidence function to include the road patch range segment total confidence.

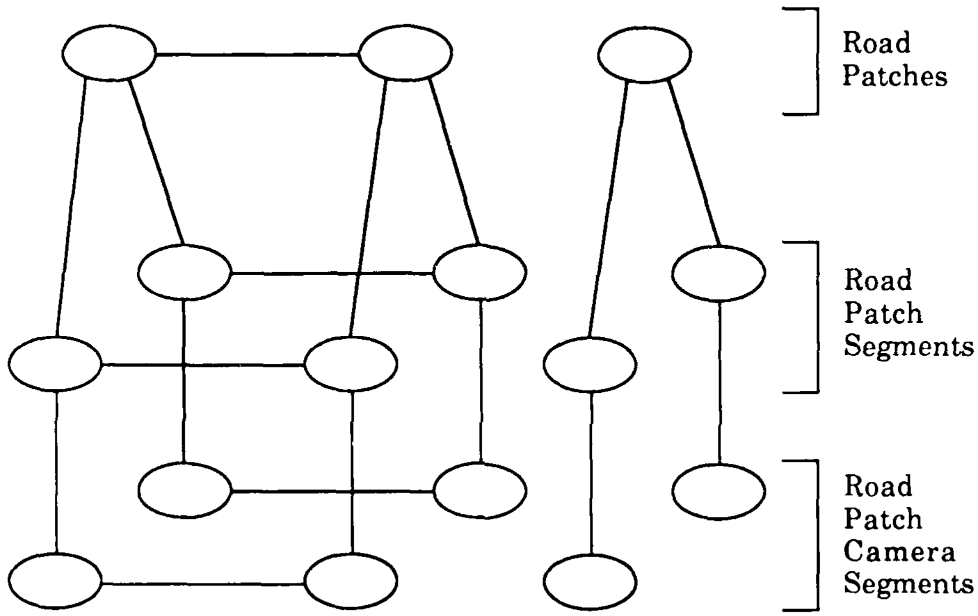


Figure 7: The Scene Model Network

5 The Scene Model Planner

The function of the ALV navigator is to examine the scene model and plot a course for the vehicle with respect to the objects. If the local navigation task is to navigate the vehicle down the center of the road, then in order for the navigator to be successful, the scene model must contain a sufficient number of relevant objects. For example, a scene model containing a series of connected road patches would be sufficient; the navigator need only plot a smooth curve between the left and right components of each road patch. However, a scene model containing houses would not provide sufficient information for the navigator to plot a path down the road; the navigator would be unsure of the proximity of the houses to the road. It is up to the Planner to decide, based on the local navigation task, what objects should appear in the scene model. However, the Planner's decision to track a particular object depends on more than the relevance of the object to the local navigation task. Choosing an object for verification should also depend on the history of tracking that object. For example, if the Planner repeatedly fails to verify a particular class of object, it should hesitate to attempt further verification; however, if verification fails due to the object being far from the vehicle, this should not prevent the Planner from attempting to verify the object if the vehicle moves closer to the object.

The Planner is implemented as a frame whose slots point to the modules with which the Planner communicates, i.e. the scene model, the a priori road map, the local navigation task, and the verifier. The unique aspect of this frame is that it inherits the capabilities of a production system, providing a rule database, a factual database, and a conflict resolution strategy. Based on the local navigation task, the a priori map, and the scene model, the production system decides what type of object to track and verify. To simplify the initial implementation of the Planner, we have assumed a constant local navigation task of following the road ad infinitum, and an a priori road map which contains the approximate locations of intersecting roads along with an approximate road width. Hence, the production system consistently selects a road patch for verification by instantiating a road patch frame whose attributes are undefined.

The next task of the production system is to choose the search location of the road patch hypothesis. The production system first queries the scene model for the directional history of the road. If the direction has varied erratically, then the Planner's confidence as to the location of the road patch is low. Imposing the constraint that the hypothesized road patch must be connected to the last road patch in the scene model improves the likelihood of verifying the road patch hypothesis. Hence, the production system defines the search strategy as "connected"; the search location is defined to be the leading edge of the connected road patch. If the road has been found to be recently straight for, say, at least 10 meters, then the Planner assumes that the road beyond the scene model is also straight. In this case the search strategy is defined to be "disconnected" and the search location is extrapolated a distance of 10 meters from the end of the scene model. If successful, the scene model can be built more rapidly in

this fashion, ultimately resulting in higher vehicle speeds.

Before the road patch hypothesis is transmitted to the Verifier, the Planner must decide which left and right features should define the road patch. This decision is based on the features defining the nearest verified road patch and the confidence with which those features were verified. The expected road width is defined as the actual road width of the nearest verified road patch in the scene model.

The Planner is now ready to send the road patch hypothesis to the Verifier, where evidence is gathered in support of it. Once complete, the Verifier returns the hypothesis to the Planner; all the attributes in the road patch frame are now defined. If the evidence is deemed acceptable by the Planner, it will add the verified object to the scene model. However, if the evidence is considered unacceptable, several options are available to the Planner:

1. hypothesize the object at a different location;
2. hypothesize a different object;
3. retain the verified components of the unverified object.

Currently, only option 1 is implemented and proceeds as follows. If the unverified road patch hypothesis is disconnected, i.e. the Planner ventured out beyond the end of the scene model to hypothesize the road patch, the hypothesis is abandoned and a connected road patch is hypothesized back at the end of the scene model. If the unverified road patch hypothesis is connected, the Planner aborts the road following task. Figure 8 summarizes the actions taken by the Planner. When additional object classes become defined, the Planner may take advantage of option 2; for example, the Planner may choose to hypothesize a ditch beside the road if it was unsuccessful in verifying the road. When different search strategies become defined, the Planner may take advantage of option 3. For example, although a road patch may not have been successfully verified, one of its component road patch segments may yield a high confidence; the Planner may choose to insert this component into the scene model.

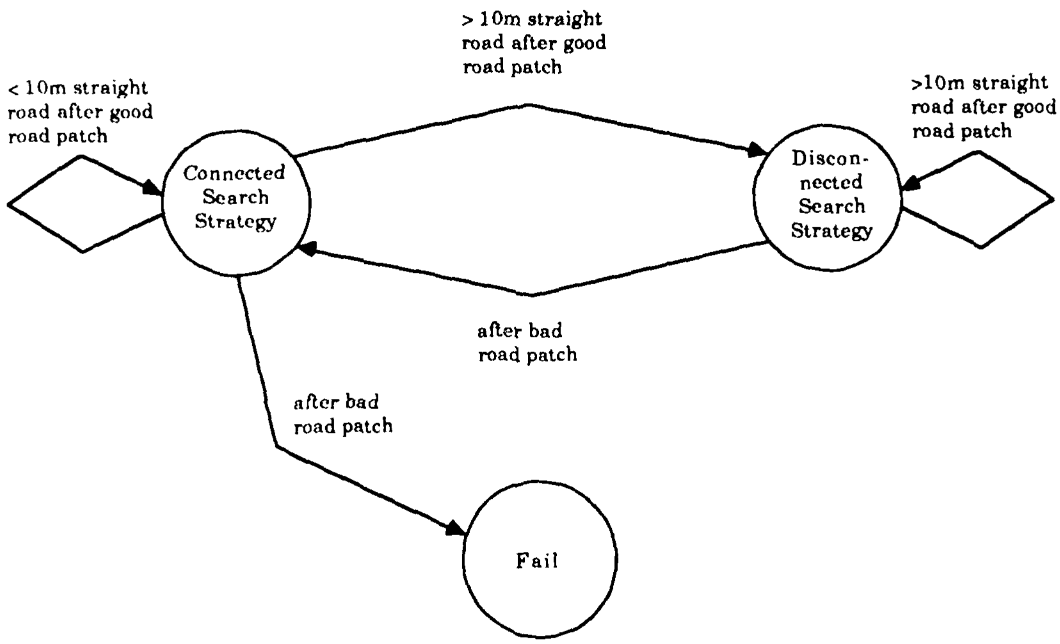


Figure 8: The Scene Model Planner States

6 The Scene Model Verifier

The role of the Verifier is to receive an object hypothesis from the Planner, collect evidence in support of the object, and return the verified object to the Planner. More specifically, when the Verifier receives an object hypothesis in the form of a sparsely defined frame, it proceeds to fill in the empty attributes; if the object has component parts, e.g. a road patch segment, the Verifier must create and define these frames. To accomplish this task, a separate blackboard has been assigned to each class of object. In Section 6.1 we describe the structure of an object blackboard; the mechanism by which blackboards communicate to verify an object is presented in Section 6.2.

6.1 The Object Blackboard

When an object hypothesis is posted on the blackboard corresponding to its class, knowledge sources are activated to fill the empty attributes of the hypothesis. As in the case of the Planner, each blackboard is implemented as a frame, providing a set of attributes and inheriting the capabilities of a production system. The attributes provide links to other blackboard frames and system modules, e.g. vehicle pilot and image processor. The production system rules control the activation of knowledge sources, i.e. when the left-hand side of a rule matches the contents of the factual database, the right-hand side activates a knowledge source. Ties are resolved by the conflict resolution strategy.

Blackboard frames, like object frames, possess both component and inheritance relationships; spatial relationships are undefined for blackboard frames. For example, the road patch blackboard has an attribute pointing to the road patch segment blackboard; although there are two component road patch segments for each road patch, there is only one road patch segment blackboard on which every instance of a road patch segment object is posted. When an object blackboard is instantiated, it may inherit the attributes and rules of other object blackboards. Consider, for example, the instantiation of a road patch blackboard; the blackboard now contains the attributes and rules of both the road patch and planar ribbon blackboard. Thus for every attribute of a road patch object, there exists one or more rules invoking knowledge sources defining that attribute. If we look carefully at the rule(s) corresponding to the *total confidence* attribute, we find that they originate in the planar ribbon blackboard, since this attribute was inherited from the the planar ribbon object. These rules control how the confidence is defined for a generic ribbon; however, this definition of *total confidence* may be inadequate in the context of a road patch. Since there is no way for the planar ribbon to anticipate which objects may inherit its attributes, it is impossible to provide a set of rules in the planar ribbon blackboard which define the *total confidence* of a road patch object. Instead, the planar ribbon blackboard retains the generic rules while the road patch provides its own rules. When the road patch blackboard inherits the planar ribbon blackboard, the redundant *total confidence* rules originating from the planar ribbon blackboard are suppressed.

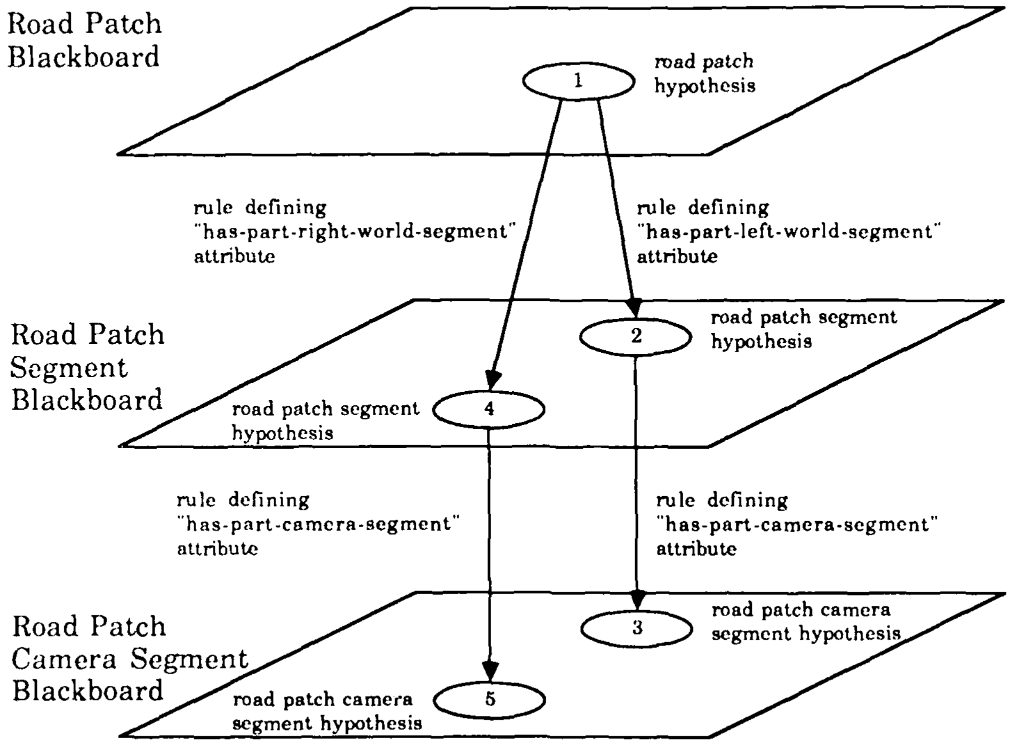


Figure 9: Top-Down Hypothesis Verification

6.2 Top-Down Hypothesis Verification

When the Planner hypothesizes an object, the Verifier invokes a top-down approach to verify the object. In Figure 9, this approach has been applied to the verification of a road patch hypothesis. Once the Planner creates the road patch hypothesis, it posts it on the road patch blackboard (a specialization of a planar ribbon blackboard). The rules belonging to the road patch blackboard, acting as daemons, invoke knowledge sources to define the attributes of the now sparsely defined road patch hypothesis. The rule antecedents ensure that the attributes are defined in a specific order. When rules fire to define the *has part left world segment* component, the activated knowledge source creates a road patch segment object hypothesis, defines a subset of its attributes, and posts it on the road patch segment blackboard. At this point, control is transferred to the road patch segment blackboard while the road patch blackboard is put to sleep.

Responding to a new object hypothesis on their blackboard, the rules belonging to the road patch segment blackboard proceed to define the attributes of the road patch segment object. When rules fire to define the *has part camera segment* attribute, the activated knowledge source creates a road patch camera segment, defines a subset of its attributes, and posts it on the road patch segment blackboard. Control is transferred to the road patch camera segment blackboard and the road patch segment blackboard is put to sleep.

The rules at the road patch camera segment blackboard proceed to define where and how the road patch camera segment is to be searched for in the camera image. When the rules fire to define *endpoint A (B)*, a knowledge source applies the *method of extraction* to the *search window* contained in the *camera image*. The results define the *endpoint A (B)* and *total confidence* attributes. At this point, all the attributes are defined and control is passed back up to the road patch segment blackboard where the remaining road patch segment hypothesis attributes are defined. Similarly, when its attributes are defined, control is passed up to the road patch blackboard. The next attribute, the *has part right world segment*, repeats the entire process until eventually control is once again at the road patch blackboard. Once the last attribute, the road patch *total confidence*, is defined, the completed road patch hypothesis is returned to the Planner for evaluation.

This system of communicating blackboards offers many advantages to the system builder. Each modular blackboard controls the definition of a single object class; as new classes are created, new blackboards are defined. If the definition of a class is changed, i.e. attributes are added or deleted, new sets of rules are added or deleted. Since rules map to single attributes, the alteration of one set of rules will have little or no impact on rules corresponding to other attributes. Within each blackboard, the inherent advantages of a rule-based system are clear. Rule-based activation of knowledge sources provides a data-driven, flexible control structure, while English-like rules provide readability and support maintainability.

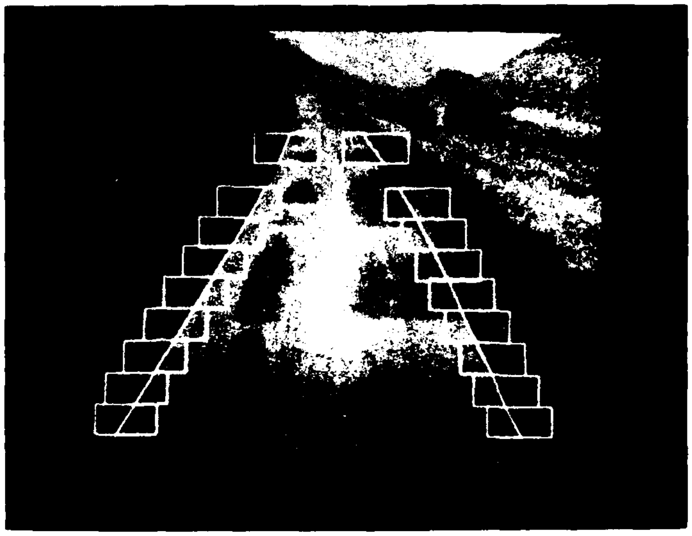


Figure 10: Tracking A Straight Road - Frame 1

7 Experimental Results

In this section we demonstrate the system on two road sequences; the road images were taken from the Martin Marietta ALV test track in Denver, CO. The current implementation runs in the Maryland Franz Lisp environment [Allen et al.], under UNIX[1] 4.3BSD on a VAX[2] 11/785. As described earlier, all system modules are frames implemented using the Maryland Franz Flavors package [Wood]; the production system frames inherited by the Planner and Verifier blackboards are implemented using YAPS [Allen]. YAPS is an antecedent-driven production system similar to OPS5 [Forgy], but offering more flexibility. Functions bound to the frames are implemented in Lisp; C routines are called from the Lisp environment for numerically-intensive processing. All image display functions are provided by a Vicom image processor.

The first sequence is shown in Figure 10 and demonstrates the construction of a scene model containing a straight road. At the bottom of the image, the initial search windows are placed according to the a priori points on the side of the road; the search windows are indicated by the rectangular boxes which contain the extracted line segments. From then on, the road patch connected search strategy is repeatedly invoked to verify successive connected road patches. Following the insertion of the eighth road patch into the scene model, over 10 meters of straight road have been

[1] UNIX is a trademark of Bell Laboratories.

[2] VAX is a trademark of Digital Equipment Corporation.

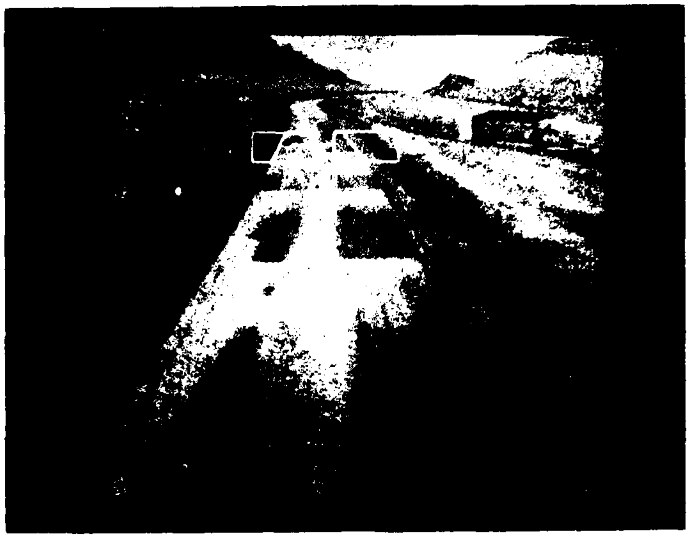


Figure 11: Tracking A Straight Road - Frame 2

accumulated. In this case, the disconnected search strategy is invoked resulting in a search location 10 meters beyond the end of the scene model. As the road was correctly predicted to be straight, the road patch is successfully verified. With approximately 20 meters of straight road in the scene model, the disconnected search strategy is again invoked; however, when the three-dimensional search location of the third road patch is mapped to the current image, the search windows are out of bounds (off the top of the image). Subsequently, as depicted in Figure 11, a new image is acquired and the windows mapped to the correct locations; again, the road patch is successfully verified.

In the second sequence, shown in Figure 12, the ALV attempts to track a curved road. As in the previous sequence, the initial portion of the road is straight; the same steps are used to build the initial eight road patches and the search strategy is changed from connected to disconnected. However, because the vehicle could not predict the upcoming curve in the road, the predicted search location is off the road, ultimately yielding a road patch with very poor total confidence (due to lack of parallelism and poor width). The Planner aborts the disconnected search strategy and invokes the connected search strategy from the previously verified road patch in the scene model. As a result, the curve is successfully navigated, as shown in Figure 13.

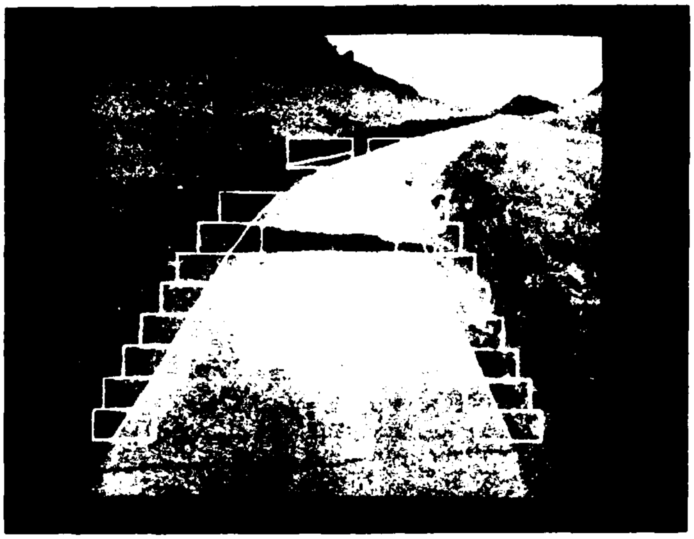


Figure 12: Tracking A Curved Road - Frame 1

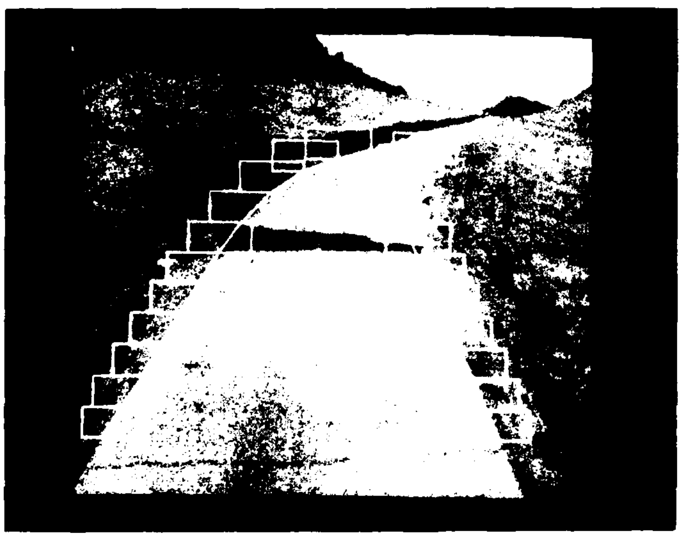


Figure 13: Tracking A Curved Road - Frame 2

8 System Evolution

In Section 7 we demonstrated two search strategies invoked by the Planner to verify a road patch. As these strategies are rather simplistic, we expect that they will evolve with time, and have designed the control structure to accommodate such evolution. In this section, we demonstrate the flexibility of the control structure by exploring the effects of altering the search strategies used by the Planner. For each proposed change, we discuss the necessary modifications to the code and present the results of running the modified system on sample images. The first two examples discuss trivial modifications and are intended to familiarize the reader with the notation; the third and final example presents a more interesting modification. While in this section we discuss modifications to the Planner, the control structure also facilitates modifications to the Verifier, as briefly discussed in Section 6.2.

8.1 Reducing the Static Road Projection

When the scene model has accumulated at least 10 meters of straight road, the Planner switches from a connected search strategy to a disconnected search strategy; the next road patch is hypothesized at a distance of 10 meters from the last road patch in the scene model. In this section, we demonstrate the changes required to alter this distance from 10 meters to 5 meters. Among the YAPS rules which collectively define the road patch *search strategy*, the following rule defines the *search strategy* as disconnected:

```
(defp define-disconnected-search-strategy
      (hypothesized object -object)
      (goal (define search strategy for road patch -object))
 test (null (<- -object 'search-strategy))
      (cond ((not (null (<- -object 'prior-road-straightness)))
             (>= (<- -object 'prior-road-straightness)
             MIN-ROAD-STRAIGHTNESS)))
  -->
      (<- -object 'set-search-strategy 'disconnected))
```

The antecedent of the rule (the four expressions preceding the "-->") is a conjunction of conditions which must be satisfied in order for the consequent (the expression following the "-->") to be executed. The first expression in the antecedent matches a road patch hypothesis created by the Planner. The second expression represents the current goal of the Planner; in this case the Planner is attempting to define the *search strategy* of the road patch hypothesis. The next two expressions, called test clauses, specify further conditions that must be met: the *search strategy* must be previously undefined and the *prior road straightness* must exceed 1000 cm (the value of MIN-ROAD-STRAIGHTNESS). The symbol, "<-", indicates message passing between objects; for example, in the first test condition, a message is passed to the road patch hypothesis, bound to the variable -object, requesting the value of the *search strategy*

attribute. If both these conditions are met, then the *search strategy* is defined as disconnected. With our new strategy, we wish to invoke the disconnected search strategy after accumulating 5 meters of straight road. Thus, the required change to the system is simply to redefine the constant, MIN-ROAD-STRAIGHTNESS, to equal 500 cm.

The next step in modifying our strategy involves the following rule which defines the *search location* of the road patch hypothesis:

```
(defp define-disconnected-search-location
      (hypothesized object -object)
      (goal (define search location for road patch -object))
 test (null (<- -object 'search-location))
      (eq (<- -object 'search-strategy) 'disconnected)
  -->
      (<- -object 'set-search-location
         (list (<- (<- *yaps-db* 'scene-model)
                   ':predict-extended-left-feature-seed
                   MAX-EXTENDED-SEARCH-DISTANCE)
               (<- (<- *yaps-db* 'scene-model)
                   ':predict-extended-right-feature-seed
                   MAX-EXTENDED-SEARCH-DISTANCE))))
```

In this rule, the consequent defines the *search location* as a value resulting from sending two queries to the scene model, requesting points extrapolated from the left and right road patch segments, respectively, of the last road patch in the scene model. To support our new strategy, the constant, MAX-EXTENDED-SEARCH-DISTANCE, must be redefined to equal 500 cm. Having altered the two constants, we obtain the results depicted in Figure 14.

8.2 Dynamic Road Projection

With our last modification, the Planner repeatedly hypothesizes the next road patch 5 meters out, provided that the last road patch was successfully verified. We now demonstrate how to make the Planner a little braver by having it move out 10 meters after the first 5 meter extension. The reasoning, of course, is that as we accumulate more straight road, we expect more to lie ahead. At some point, however, the verifier has trouble verifying road patches that are too far ahead; we choose 10 meters as our limit. To accommodate this dynamic projection search strategy, we add the following YAPS rule to the Planner:

```
(defp define-dynamically-disconnected-search-strategy
      (hypothesized object -object)
      (goal (define search strategy for road patch -object))
 test (null (<- -object 'search-strategy))
      (cond ((not (null (<- -object 'prior-road-straightness)))
             (>= (<- -object 'prior-road-straightness)
```

Figure 14: Reducing the Static Road Projection

```
                MIN-ROAD-STRAIGHTNESS)))
      (eq (<- (<- (<- *yaps-db* 'scene-model)
                ':retrieve-most-recent-road-patch) 'search-strategy)
           'disconnected)
  -->
      (<- -object 'set-search-strategy 'dynamically-disconnected))
```

In the test clauses, we check that the *search strategy* of the road patch hypothesis is undefined and make sure that we have accumulated sufficient straight road in the scene model. In addition, we check that the previously verified road patch was verified using the disconnected search strategy. If all these conditions hold, the *search strategy* is defined to be dynamically disconnected.

To define the *search location* for this new strategy, we add the following YAPS rule:

```
(defp define-dynamically-disconnected-search-location
      (hypothesized object -object)
      (goal (define search location for road patch -object))
 test (null (<- -object 'search-location))
      (eq (<- -object 'search-strategy) 'dynamically-disconnected)
  -->
      (<- -object 'set-search-location
         (list (<- (<- *yaps-db* 'scene-model)
                   ':predict-extended-left-feature-seed
```

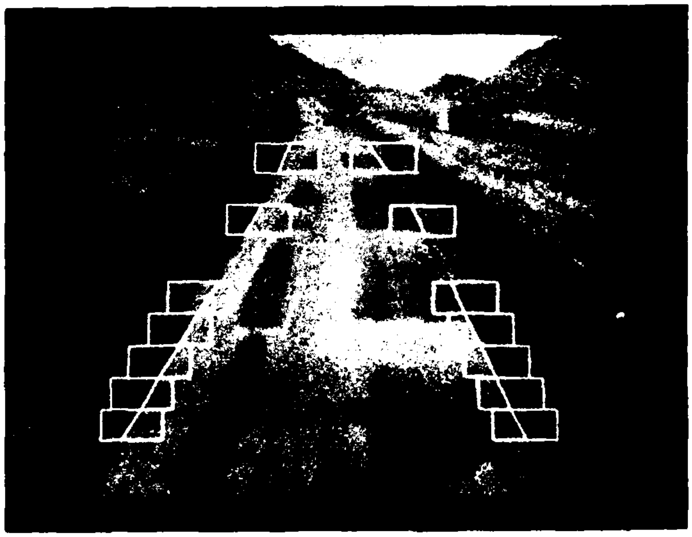


Figure 15: Dynamically Increasing Road Projection

```
      MAX-EXTENDED-SEARCH-DISTANCE)
   (<- (<- *yaps-db* 'scene-model)
      ':predict-extended-right-feature-seed
      MAX-EXTENDED-SEARCH-DISTANCE))))
```

In this rule, the rule consequent defines the *search location* to be 10 meters (the value of MAX-EXTENDED-SEARCH-DISTANCE) from the last road patch in the scene model. The results of this new strategy are depicted in Figure 15.

8.3 Dynamically Decreasing Road Projection

Returning to our original search strategy, the Planner aborts the disconnected search strategy in the event of an unverifiable road patch hypothesis. Rather than returning to the connected search strategy, we might try to relax our projected search location and re-hypothesize the road patch halfway between the unverified road patch and the last verified road patch in the scene model. We have as our original rule:

```
(defp road-patch-disconnected-to-connected-retry
      (hypothesized object -object)
      (goal (process road patch hypothesis -object))
 test (eq (<- -object 'search-strategy) 'disconnected)
      (cond ((not (null (<- -object 'total-confidence)))
             (< (<- -object 'total-confidence)
```

```
              MIN-ROAD-PATCH-CONFIDENCE)))
  -->
      (<- -object 'set-search-strategy 'connected)
      (<- -object 'set-back-connected-planar-ribbon
         (<- (<- *yaps-db* 'scene-model)
            ':retrieve-most-recent-road-patch))
      (<- -object 'set-search-location
         (list (<- (<- (<- -object 'back-connected-planar-ribbon)
                  'has-part-left-world-segment) 'endpoint-B)
               (<- (<- (<- -object 'back-connected-planar-ribbon)
                  'has-part-right-world-segment) 'endpoint-B)))
      (<- -object 'set-has-part-left-world-segment nil)
      (<- -object 'set-has-part-right-world-segment nil)
      (<- -object 'set-actual-width nil)
      (<- -object 'set-actual-width-confidence nil)
      (<- -object 'set-parallelism-confidence nil)
      (<- -object 'set-total-confidence nil))
```

The antecedent of this rule checks that the confidence of the road patch was too low, i.e. that the road patch was unverified, and that the disconnected search strategy was invoked to verify the road patch. The consequent of the rule clears the contents of the road patch hypothesis, resets the *search strategy* to be connected, sets the *back connected planar ribbon* to the last road patch in the scene model and sets the *search location* to the end of the scene model. Our new search strategy results in the following rule:

```
(defp disconnected-to-halfway-retry
      (hypothesized object -object)
      (goal (process road patch hypothesis -object))
 test (eq (<- -object 'search-strategy) 'disconnected)
      (cond ((not (null (<- -object 'total-confidence)))
             (< (<- -object 'total-confidence)
                MIN-ROAD-PATCH-CONFIDENCE)))
  -->
      (<- -object 'set-search-location
         (<- -object 'set-search-location (halfway
            (<- (<- (<- *yaps-db* 'scene-model)
               ':retrieve-most-recent-road-patch)
               'search-location)
            (<- -object 'search-location))))
      (<- -object 'set-has-part-left-world-segment nil)
      (<- -object 'set-has-part-right-world-segment nil)
      (<- -object 'set-actual-width nil)
      (<- -object 'set-actual-width-confidence nil)
```

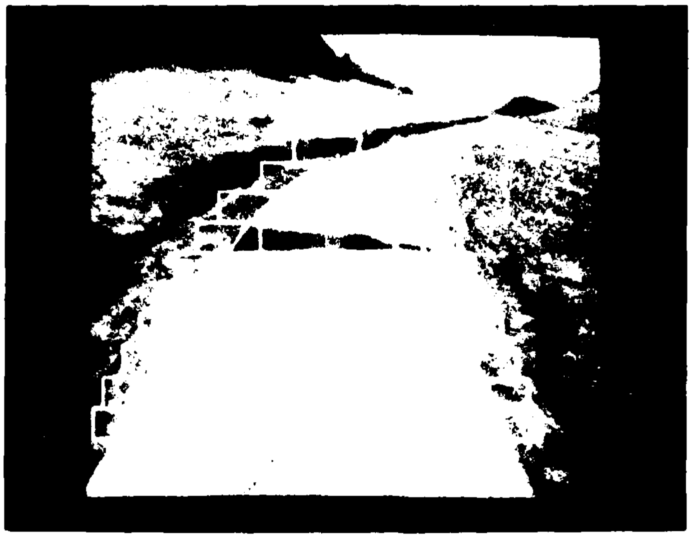


Figure 16: Dynamically Decreasing Road Projection

```
(<- -object 'set-parallelism-confidence nil)
(<- -object 'set-total-confidence nil))
```

In this case, the *search location* is calculated by a function finding the midpoint between the last road patch in the scene model, and the unverified road patch. The new rule bears close resemblance to the original rule; we have simply removed the first two components of the rule consequent, and have provided a new definition of the search location. Applying this strategy to the unverifiable road patch in Figure 12, we obtain the results depicted in Figure 16.

9 Related Work

The decomposition of an object both by component and by level of abstraction, and the construction of hierarchical frame networks, bear close resemblance to techniques used in the VISIONS system [Hanson & Riseman]. In the VISIONS system, the long term memory (LTM) contains a priori visual knowledge of the world, while the short term memory (STM) represents the interpretation of the scene. Both the LTM and STM are structured as a hierarchy of levels of representation defining the levels of object abstraction. The control strategy first decides which partial model (frame network) to focus on, expands (hypothesizes) a node, and finally verifies the node. Although originally defined for outdoor house scenes, this work has recently been extended to the road following task [Arkin et al.].

[Lawton et al.] describe a system also resembling the VISIONS system. The short term memory (STM) acts as a dynamic scratchpad for the vision system, containing object hypotheses, incoming imagery, and the results of feature extraction. When hypotheses accumulate sufficient evidence, they are moved to the long term memory (LTM), which includes a priori terrain representations. The control structure provides both top-down and bottom-up hypothesis instantiation over the network hierarchies.

Although hypothesis instantiation in the above systems is both top-down and bottom-up, the entire sensor image is processed to initialize the short term memory with image features; local image processing in our system is based on [Le Moigne et al.]. In addition, none of these systems use a rule-based production system to invoke knowledge sources.

[Smith & Strat] describe an information manager that is the core of a sensor-based autonomous system. A centralized knowledge database is proposed, accessible to a community of independent asynchronous processes. The representation scheme organizes data tokens in both an octree and a semantic network thus supporting both spatial and semantic queries. The independent asynchronous processes can be activated by daemons embedded in the database or by procedure call.

10 Conclusions

The system described in this report provides a flexible architecture for constructing an ALV scene model. The representation of objects as networks of frames offers a natural grouping of knowledge; the multiple layers of abstraction facilitate the addition of new sensor features in support of existing world objects. Construction of the frame network is provided by a set of modular blackboards providing top-down instantiation of the frames in the network. Each blackboard, implemented by a production system, is an "expert" in defining a particular class of frame; the English-like rules governing the invocation of knowledge sources are easy to understand, and narrow the gap between control specification and implementation. From a system maintenance standpoint, all object frames, blackboards, production system tools, and object oriented programming tools are off-the-shelf; these facilities are documented, tested, and readily accessible. The implementation languages supporting the system cover the needs of the programmer; YAPS offers high-level encoding of control strategy, Lisp provides symbolic manipulation, C speeds up numerical processing, and Flavors facilitates inter-object communication.

The system is currently being expanded to support new planning and verification strategies. The Planner is being supplemented with strategies for road following in the event that a connected road patch cannot be verified. This includes proceeding past an unverified road patch provided that it contains a verified component, and invoking an exhaustive search for road patches in a given area; the latter strategy will be accomplished using bottom-up verification in which road patch camera segments posted at lower levels generate instances of road patches at upper levels. This integration of top-down and bottom-up verification will remove some of the burden placed on the Planner of accurately predicting the location of an object.

References

[Allen et al.] Allen, Elizabeth M., Randall H. Trigg, and Richard J. Wood, "The Maryland Artificial Intelligence Group Franz Lisp Environment, Variation 2", Technical Report TR-1226, Department of Computer Science, University of Maryland, November 1983.

[Allen] Allen, Elizabeth M., "YAPS: Yet Another Production System", Technical Report TR-1146, Department of Computer Science, University of Maryland, December 1983.

[Arkin et al.] Arkin, Ronald C., Edward M. Riseman, and Allen R. Hanson, "AuRA: An Architecture for Vision-Based Robot Navigation", Proceedings: 1987 DARPA Image Understanding Workshop, pp. 417-431, Los Angeles, CA, February 1987.

[Duda & Hart] Duda, Richard O. and Peter E. Hart, "Pattern Classification and Scene Analysis", pp. 393-396, John Wiley & Sons, Inc., 1973.

[Forgy] Forgy, C. L., "OPS5 User's Manual", Technical Report CMU-CS-81-135, Department of Computer Science, Carnegie-Mellon University, July 1981.

[Hanson & Riseman] Hanson, Allen R. and Edward M. Riseman, "VISIONS: A Computer System for Interpreting Scenes", in Hanson, Allen R. and Edward M. Riseman, (eds.), "Computer Vision Systems", Academic Press, Inc., 1978.

[Lawton et al.] Lawton, Daryl T., Tod S. Levitt, Christopher C. McConnell, Philip C. Nelson, and Jay Glicksman, "Environmental Modeling and Recognition for an Autonomous Land Vehicle", Proceedings: 1987 DARPA Image Understanding Workshop, pp. 107-121, Los Angeles, CA, February 1987.

[Le Moigne et al.] Le Moigne, Jacqueline, Allen M. Waxman, Babu Srinivasan, and Matti Pietikainen, "Image Processing for Visual Navigation of Roadways", Technical Report CAR-TR-138, Center for Automation Research, University of Maryland, July 1985.

[Minsky] Minsky, Marvin, "A Framework for Representing Knowledge", in Patrick Henry Winston (ed.), "The Psychology of Computer Vision", McGraw-Hill, Inc., 1975.

[Smith & Strat] Smith, Grahame B., and Thomas M. Strat, "Information Management in a Sensor-based Autonomous System", Proceedings: 1987 DARPA Image Understanding Workshop, pp. 170-177, Los Angeles, CA, February 1987.

[Srinivasan] Srinivasan, Babu, "Image Processing Algorithms for Navigating Roadways", MS Thesis, University of Maryland, 1984.

[Wood] Wood, Richard J., "Franz Flavors: An Implementation of Abstract Data Types in an Applicative Language", Technical Report TR-1174, Department of Computer Science, University of Maryland, June 1982.

A Object Model Attributes

A.1 The Road Patch

The following attributes comprise the road patch frame:

- **search location:** The search location specifies the expected location of the road patch in terms of a three-dimensional line segment. The orientation of this segment, called a rib, is such that the left and right boundaries of the road are expected to pass through the left and right endpoints, respectively; hence, the direction of the line segment is perpendicular to the expected direction of the road in that vicinity. Figure 4 shows the search location for the next road patch.

- **search strategy:** The search strategy specifies the manner in which the road patch is to be verified. It is assumed that the vehicle is initially located on an a priori verified road patch. Road Patch 1 in Figure 4 is the initial road patch. From that point on, each road patch is verified according to one of the following strategies:

    1. *connected:* The connected search strategy is used to verify a road patch which is connected in the front and/or the back to another road patch. Road Patches 2, 3, and 4 in Figure 4 are examples of connected road patches.
    2. *disconnected:* The disconnected search strategy is used to verify a road patch which is connected in neither the front nor the back to another road patch. Road Patch 5 in Figure 4 is a disconnected road patch.

- **front (back) connected planar ribbons:** If the road patch is front (back) connected to the back (front) of another road patch, this attribute points to the connected road patch.

- **has part left (right) world segment:** This attribute points to the frame representing the left (right) road patch segment, which is a boundary of the road.

- **expected width:** The expected width is based on the actual width of the road patch in closest proximity.

- **actual width:** The actual width is defined as the average perpendicular distance from the endpoints of the left road patch segment to the ray defined by the right road patch segment. This is illustrated in Figure 17.

- **actual width confidence:** The actual width confidence is defined as follows:

$$\max(0.0, 1.0 - \frac{|ActualWidth - ExpectedWidth|}{ExpectedWidth})$$

Intuitively, the fractional term represents the deviation of the actual width from the expected width.

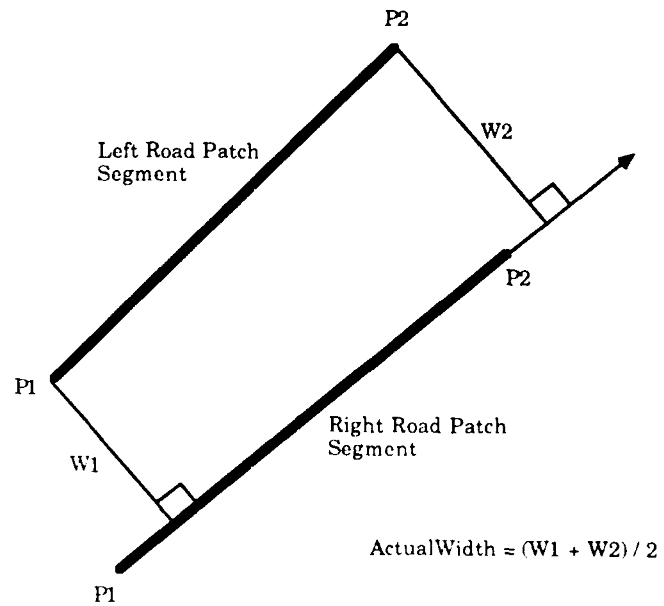


Figure 17: The Calculation of Actual Width Confidence

- **parallelism confidence:** The parallelism confidence is defined as follows:

$$\frac{90 - \theta}{90}$$

where θ is the acute angle between the left and right road patch segments. The maximum confidence is achieved when the two segments are parallel, while minimum confidence is achieved when the segments are orthogonal.

- **total confidence:** The total confidence is a function of the width confidence, the parallelism confidence, and the total confidence of the left and right component features:

$$\begin{gathered} 0.40 ParallelismConfidence + \\ 0.20 WidthConfidence + \\ 0.20 LeftRoadPatchSegmentTotalConfidence + \\ 0.20 RightRoadPatchSegmentTotalConfidence \end{gathered}$$

The total confidence function has been empirically determined and gives added weight to road patches whose left and right road patch segments are parallel.

- **prior road straightness:** The prior road straightness specifies the length (in meters) of straight road from the front of the last verified road patch extending back into the scene model. The prior road straightness of the hypothesized patch in Figure 4 is given by the arrow.

- **left (right) world segment type:** The left (right) world segment type specifies whether the component left (right) road patch segment defines the edge between the road surface and the road shoulder surface (road edge), or the edge between the road shoulder and the vegetation or background (shoulder edge).

A.2 The Road Patch Segment

The following attributes comprise the road patch segment frame:

- **search location:** The search location specifies the expected location of the road patch segment in terms of a three-dimensional point through which the road patch segment is expected to pass. This point is one of the endpoints belonging to the search location of its parent road patch.

- **search strategy:** The search strategy specifies the manner in which the road patch segment is to be verified. Each road patch segment is verified according to one of the following strategies:

  1. *connected:* The connected search strategy is used to verify a road patch segment which is connected in the front and/or the back to another road patch segment.

2. *disconnected:* The disconnected search strategy is used to verify a road patch segment which is connected in neither the front nor the back to another road patch segment.

- **endpoint A (B):** Endpoint A specifies the three-dimensional coordinates of endpoint A (B). It is calculated by applying a flat earth inverse perspective transformation [Duda & Hart] to endpoint A (B) of its component road patch camera segment frame.

- **endpoint A (B) connected world segment:** If the road patch segment is connected at endpoint A (B) to another road patch segment, then this attribute points to the connected road patch segment.

- **part of planar ribbon:** This attribute points to the parent road patch frame.

- **has part camera segment:** This attribute points to the frame representing the component road patch camera segment.

- **total confidence:** The total confidence is a function of the continuity confidence (defined below) and the total confidence of its component road patch camera segment. The total confidence is calculated as follows:

$$0.30 ContinuityConfidence +$$
$$0.70 RoadPatchSegmentTotalConfidence$$

- **world segment feature:** The world segment feature specifies whether the road patch segment defines the edge between the road surface and the road shoulder surface (road edge), or the edge between the road shoulder and the vegetation or background (shoulder edge).

- **endpoint A (B) orientation:** This attribute defines the orientation of endpoint A (B) as front or back.

- **continuity confidence:** The continuity confidence, defined only when the road patch segment is connected, measures the degree to which the road patch segment and its connected neighbor lie on the same vector. The continuity confidence is calculated as follows:

$$1 - \frac{180 - \theta}{180}$$

where θ is the angle between the two road patch segments.

A.3 The Road Patch Camera Segment

The following attributes comprise the road patch camera segment frame:

- **search window:** The search window specifies the expected location of the road patch camera segment in terms of a two-dimensional rectangular window defined over a camera image. If the road patch camera segment is to be verified using the connected search strategy, the window is located adjacent to that from which the connected road patch camera segment was extracted. However, if the road patch camera segment is to be verified using the disconnected search strategy, the window is centered around the point defined by applying a flat earth direct perspective transformation [Duda & Hart] to the search location of its parent road patch segment.

- **search strategy:** The search strategy specifies the manner in which the road patch camera segment is to be verified. Each road patch camera segment is verified according to one of the following strategies:

  1. *connected:* The connected search strategy is used to verify a road patch camera segment which is connected in the front and/or back to another road patch segment.

  2. *disconnected:* The disconnected search strategy is used to verify a road patch camera segment which is connected in neither the front nor the back to another road patch camera segment.

- **endpoint A (B):** The two endpoints are the result of applying a linear feature extractor to the search window. The endpoints are constrained to lie on the border of the search window and define a two-dimensional line segment.

- **endpoint A (B) connected camera segment:** If the road patch camera segment is connected at endpoint A (B) to another road patch camera segment, then this attribute points to the connected road patch camera segment.

- **part of world segment:** This attribute points to the parent road patch segment frame.

- **camera image:** This attribute points to the camera image on which the search window is defined.

- **method of extraction:** The method of linear feature extraction depends on the search strategy. If the search strategy is connected, a method based on a one-dimensional Hough transform is applied to the search window using the connected endpoint as a pivot. If the search strategy is disconnected, a method based on a two-dimensional Hough transform is applied to the search window. These methods are illustrated in more detail in [Srinivasan].

- **total confidence:** The total confidence ranges from 0 to 1 and is based on the strength of the extracted linear feature.

- **camera segment feature:** The camera segment feature specifies whether the road patch camera segment defines the edge between the road surface and the road shoulder surface (road edge), or the edge between the road shoulder and the vegetation or background (shoulder edge).
- **endpoint A (B) orientation:** This attribute defines the orientation of endpoint A (B) as front or back.

END
DATE
FILMED
4-88
DTIC